

# ATARI COMPUTER ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

MAY, 1986

Mike Dunn, Jim Bumpas, Larry Gold, co-editors

Task Game Play Score Options

Hide letter values  
Show distribution  
Hide computer move  
Set skill level...  
Set time limit...  
Sound on

Jim's letters  
G A T E I O

Hide Shuffle

1. Type your word:  
GATE

2. Which direction?  
⬅ ➡

3. Click on the square where the word starts.



## FLASH COMDEX REPORT

**ATARI:** Buy a 520 ST and get a 2" drive for \$70. 1040 STs have socket for blitter chip; older models will have add-on board. No heat problems are reported with all-in-one box. Only 30% of ST sales are in the US. The ST is the biggest selling computer in Germany. The 20-meg hard disk is \$800; the SMM804, 80cps dot printer is \$219. New ST 3<sup>rd</sup> party catalog is 347 pages.

**ANTIC:** STart, ST quarterly. 8-Bit: Calc Magic, a \$25 spreadsheet for the 130XE. ST: CAD-3D \$50 by Tom Hudson; A-Calc, a \$60 GEM spreadsheet; Expert Opinion, \$100 professional AI expert system from France.

**BATTERIES INCLUDED:** 8-Bit: advanced Paper Clip with spellpack for the 130XE. ST: versions of PaperClip and HomePak; Degas Elite and Isgur Portfolio analyzer. The sensational Thunder, an advanced real-time spelling checker for \$40.

**ABACUS:** PC board designer for \$400, plus all their manuals (5 ready and more to come, including the definitive ST Encyclopedia).

**MINDMINE:** 520 ST memory upgrades, \$50 real time clock, and a customized to your picture ST-talking doll for only \$150.

**PCA:** Wonderful \$500 program for desktop publishing with laser printer or Epson. It's also a CAD system, word processor, graphics modelling spreadsheet, macros, etc. Potential for being the most significant product shown.

**MICHTON:** Many new utilities, and one of the few showing ST games.

**TLM:** OS/9 68k multiuser/multitasking "unix like" software which will run GEM based non-graphic software and any OS/9 software.

**XLENT:** 8-Bit: PS Interface, to link Printshop, Xlent's graphic programs, & a new powerful \$30 word processor.

**GSI** had a very fast fractal generator with powerful zoom functions.

**Hippo** (\$140) and Print-Technik (\$250) both showed video digitizers. Hippo also showed eprom burners, idea processor, audio sampler, and its complete line.

**HABA** showed a \$99 memory upgrade for the 520 ST, a 1200 bps modem, and a powerful program which uses all the protocols, including xmodem, Compuserve & Kermit. Powerful business applications were shown by Regent, Power Systems, Sierra On-Line, Business Operating Software, DAC, Versasoft & others..

**Hybrid Arts** showed a MIDI synthesizer for the 8-bits in the \$40-\$60 range. Activision showed its electronic music publishing software.

Languages for the ST include at least 3 Basics (one compiled), several C versions (OSS, Mark Williams, Lattice, etc.). TDI showed Module-2, Fortran, Cobol, Forth.

**Quantum** showed a product using the MIDI port (\$149) which permits up to 20 STs to share peripherals. Paul Heckel of QuickView was there with his Zoomracks and his Friendly Software Design book.

The best part was the wonderful Antic 4<sup>th</sup> anniversary awards party where I accepted the first Antic User Group Award on behalf of all of you. Many of the vendors were ACE members before they became professional, and I met many old and new friends. The Antic award ceremony was well attended and very nice. Everyone had a good time and the awards are beautiful. It was a real honor to accept the award on behalf of all you ACE members!

-- Mike Dunn

## STuff

The 520 ST is now for sale at Toys R Us. The new 520s and the 1040s may use a composite monitor (and TV?). The 20-meg hard drive is just now going into serial production. You should see them soon. ST Basic is being revised, and improvements in the GEM tools are also scheduled. The Amy sound chip may also be on its way back onto Atari drawing boards.

Hitachi CD-ROM drive for \$680 is now selling for the ST by Liberty Corp., in Washington state.

Datapac Inc. plans to distribute a ROM cartridge for the ST which will permit the user to run Macintosh programs.

Shanner is now selling a dual, double-sided, disk drive for \$399. Find someone who wants your single-sided drive and have 2 double-sided drives for \$200 less than the price of 2 separate drives.

Antic On-Line quotes Sam Tramiel regarding 8-bit developments coming later this year, including: A plug-in 80-column card with a parallel printer interface for \$79; memory chip expansions similar to those Apple has for the IIe; 500K memory 3.5-inch disk drives, with a new DOS by OSS.

## ST Library

Thanks to Andre Lafreniere and all our friends in Quebec we have some dramatic new additions to our disk library. Some of this material is from Germany. Some of the documentation and print statements in the program code is written in French and German. The following disks are added:

1. Midi Write Program disk;
2. Telecommunications disk (including a patch to the MITERM program, and a Super BBS);
3. Digitized photos;
4. a disk full of fonts and two font loader desktop accessories;
5. a disk full of basic programs.

We've also split off the C files from the Utility disk and have created a special C disk, adding many new utilities to the Utility disk. These utilities include: Filecmp.prg; Speed.prg (check your floppy speed); Volume.prg and .rsc (change volume names on disk); Copy3.acc (a desktop sector copy utility); STscope.prg (sector editor); Picsw.prg and .txt (converts MacPaint! and C-Amiga .IFF! files to Degas or NeoChrome format — now we have access to all the art available for the 68k systems!); RDspeed.tos (test your ramdisk speed).

Other new programs include: Mandel.prg; Compose.prg (a demo of a new product soon to appear from Xlent Software); Celest.prg (Celestial Caesars strategy game); and Life.prg.

One spectacular item we've added to our MIDI & Sound demo disk is a 25k file of digitized sound. It sounds like a rather badly recorded rock and roll song off the radio. But it's a start. Also we've been receiving quite a few games. We have almost a disk full of games. If you want the latest printout of our library list, send 50 cents. Each single-sided disk we ask a \$10 donation; \$15 for a double-sided disk.

## MEETING

### WEDNESDAY MAY 14TH

### 7:30 PM

### SOUTH EUGENE HIGH

### SOMETHING SPECIAL



# BUMPAS REVIEWS

## WORD FOR WORD

**Word for Word** (\$40, BayView Software, 177 Webster Street, suite A-295, Monterey, CA 93940) is the game of Scrabble for your ST. Any number of the one to four players may be humans or the computer. The program is also available for IBM PCs and compatibles which use the GEM interface. The game will play on either a color or monochrome monitor.

This game is really much more than the ordinary game of Scrabble. How easy is it for you to design and use a custom board with your old Scrabble game? WFW gives you three additional boards from which to choose, as well as a utility with which you can design boards of your own. You can alter the size, shape, and layout of the board. A different board is selected by pulling down a menu from the desktop from within the game. WFW also keeps score for the players and will also search its own 20,000 word dictionary for words to assist a player in selecting a high-scoring word.

One of three skill levels can be chosen, and time limits may be set. Word may be challenged, after which the players look up the word in a reference of choice. If the decision is "not a word", then the offending player loses that turn. Play options include passing a turn and exchanging letter tiles. The values of letters may be displayed. A player may "hide" his letters during his turn to keep the other players from seeing them.

The game is extremely easy to play, with most game functions being executed with the mouse. All one's desktop accessories are also available along with the various other drop-down menus. I highly recommend this program for anyone who enjoys Scrabble or word games.

## ST-TERM

**ST-TERM** (\$30, Commnet Systems, 7348 Green Oak Terrace, Lanham, MD 20706) does most of what PC/Intercomm does, but for a much more reasonable price. I was not able to invoke any VT100 emulation (even though the magazine ads I've seen for it say it has "VT52/VT100" emulation), but it will emulate the 8-bit Ataris. And it has several features missing (can you believe it?) from PC/Intercomm.

For instance: You have access to 9 DOS functions. You can actually perform file maintenance functions on the disk while in the program. It also contains a connect timer/billing calculator.

Kermit and Xmodem protocols are supported and there are 20 definable macro function keys. The documentation and program also provide for saving particular setups (bps rate, etc.) for each number you might call on the autodial directory. But I was unable to get this feature to work. Everytime I used Alt-S, I got the message telling me the file was written with "0 bytes". Even after a phone conversation with the program's designer, I was still unable to get any different result. The program is supposedly no longer protected, but I was told the problem might be occurring because I was trying to save setup files to a backup disk. So I believe this is a problem which needs correction, or at least better information to users so they may correctly save their setup files.

There are 20 pages of documentation which are complete, but lack either a table of contents or index. You'll probably not be able to remember how to use all the features until you become very familiar with the program, so a quick way to get to relevant sections might be nice.

This program is brought to you by the same folks who give you FoReM ST, and this terminal software is designed to work with that BBS software. Full compatibility if promised. This program looks like a good bet for those of you who want most of what PC/Intercomm offers, and who do not need the VT100 emulation.

## HINTS

### For New Users

M.U.L.E. players will be interested to know I have been informed by a local ACE member who says he's scored over 900,000 points in a game. I tried his method and got over 231,000 my first try. Here's how you do it:

You must have at least 2 human players. Try to build up stocks of crystite (and smithore) so each player has an accumulation of 50 units of these ores on Turn 11. Try to produce maximum crystite on Turn 12. During the auction on Turn 12, make sure at least one player is on the "Buy" line for crystite. Run the buying player up as high as you can (you should be able to get to over \$800 easily). At the very last moment, drop a seller down and sell at least one unit. This will make all the 400 or so crystites accumulated among the players worth more than \$800 each for the final accounting. Buy one unit of all the other commodities also, just to drive the value of the colony up by the increment.

## COPYRIGHTS

We're seeing more and more software appearing in the public domain these days. And the quality is very good. Many of these new programs carry what I consider a misleading "copyright" notice. I think the authors of the software might be the major victims of this misunderstanding. The notice usually reads something like the following: "copyright 1986 by Software Developer, all commercial rights reserved. This software may be freely copied and distributed, but it may not be sold...."

The law requires the copyright holder to protect his rights. Once one gives anyone the right to copy and distribute the software, it is in the public domain. I don't believe one can require users to protect one's "commercial" rights after one has given the material to the public domain. Once in the public domain, a user may make any use of the software desired: Copying, distributing, altering, even selling. There are several companies which make a business of collecting good public domain software and re-issuing it in their product line for a profit. This is perfectly legal. I do not believe someone who has ostensibly retained "commercial rights" will be able to prosecute a claim for copyright infringement after the software has been placed in the public domain.

If you are a software developer, and you believe you have a good piece of software which you might want to distribute commercially, don't be misled. You may still distribute it at a profit, of course. But I don't believe you'll be able to legally prevent anyone else from doing the same. You might be well advised to spend \$25 for a half hour consultation with an attorney to discuss the issues I've raised here. But software developers are having a hard enough time in today's market without thinking you're protecting rights when you're not. Look into it.

— Jim Bumpas

## SCREENS

**SCREENS** (by Joseph J. Wrobel, \$20 from The Soft Cellar, P. O. Box 16393, Rochester, NY 14616-0393) is a lovely hack for all you 8-bit programmers to use and enjoy (and amaze your ST-owning friends).

WINDOWS, on your trusty old 400, 800 or XL! 9 of them at once, displaying messages, graphs, pictures, time, the disk directory, whatever — all without interfering with what is otherwise on the display.

**SCREENS** is a relocatable machine-language utility which loads, as an AUTORUN.SYS file, just above your DOS in RAM, protecting itself by moving MEMLO up by 3297 bytes. Once there, it gives you a new device, "W:", with a simple series of BASIC commands to specify the location and size of windows 1-9, the position, color, and size of the display inside the window, and the choice of two ways of framing the window, if desired.

I am delighted to learn I can display any character in ROM, and/or custom characters, or the most complex graphics, in any size — from a single pixel (!) to a full screen — by simply specifying two numbers, the width and height of characters in a window. And I can move, re-size, and superimpose windows, store them in RAM or on disk, recall them, reverse their colors, and even scroll their contents.

The commands for doing all this are simple BASIC ones which can be included in any program. Examples are given in the 36 page manual and on the (unprotected) disk. To get you going there is an impressive DEMO program, with other, smaller programs which make it easy to get started with this flexible utility. Once you learn how to work the "W:" device — think of it as a second TV screen, like the "S:" device, or a kind of printer, "P:", with its own special but essentially simple rules, the only limit (aside from BASIC's leisurely ways) is your imagination and programming skill.

Are you up to the challenge? Then send your \$19.95 to the Soft Cellar.

You'll get a disk with the device handler program (as an AUTORUN.SYS file), four custom character fonts, five BASIC demo programs (with handy subroutines which you can lift bodily for your own programs), a delightful picture of an Atari 800, and the manual. This modest but complete booklet gives you a 21-page tutorial, a handy reference section with details of the terms and commands used, and a chapter for advanced programmers which details how "SCREENS" loads itself into RAM, the OS variables and page zero locations it uses (without leaving bugs behind for other programs), and the formats used to save and recall a window in RAM or on disk. Some of you might be glad to know that Page 6 is NOT used.

**SCREENS** can, in principle, be used with other languages which are faster than BASIC, if you know how. A hint: You can load and recall windows from any address up to 65535.

Now, let's see, by reading the joystick port or console and poking PORTB I could....

— Dick Barkley



## News and Reviews

Mike Dunn, Co-Editor

Last month, two significant events happened to ACE. We were named by ANTIC as the best Atari Club, and we almost ran out of money. I will be going to COMDEX in Atlanta to receive the award and hopefully report in this issue on the show. The last I heard, Atari will have at least 17 smaller companies in their booth showing their new products, so it should be interesting. The money issue is more complicated. We have just developed a fantastic BBS using a super 130xe Atari with 512K memory, a 10 Meg hard disk, a 300/1200/2400 bps modem, and a fabulous new software system — all of which cost us lots of money to develop. Usually we keep about 2 to 3 issues ahead in the bank, but we got a little low. The other thing that happened is we received several large orders for software from foreign companies, and our bank would not accept them directly — they needed to be sent back to the country of origin before they would credit them. After our plea, we recieved a welcome influx in cash from you, our members, including several generous gifts for the BBS — thank you! We also invite you to buy some of our disks from the library — this will help alot.

I am frequently asked by new members and new clubs on how to make a club successful. I am sure there are many ways, but I want to tell you about our way. When we started the club 6 years ago, we decided on certain principles and have not changed them.

1. The club is open to anyone; there are no dues, but the newsletter needs to be paid for if you want it mailed.

2. Since we are a small town, we try to help out others living in small towns where there are no other users around or no club. When I had the only Atari in Eugene, I could understand how much help, even long distance help, was useful.

3. We try to have members be as helpful as they wished, no matter where they lived. Stan Ockers lives in Illinois, Sydney Brown in Australia, George Suetsugu in Hawaii; all are active members although none have ever attended a meeting.

4. The business meetings are separate from our general meetings and are usually attended only by the "Council", that is, the officers and other active members but all are welcome. Most people do not want to spend their time discussing the general business of running the club.

5. There are a variety of types of meetings — guest speakers, help sessions, demos of new products, swap meets, etc.

6. There are a variety of special interest groups meeting seperately. In the past these have included hardware groups, educational groups, and special language groups, including BASIC, ACTION!, assembly language, LOGO, PILOT, and now C and ST.

7. Everything we do is public domain, and allow you to do anything with them except to sell them commercially. Other clubs can sell our software, but only one company (Computer Palace) can sell them. Others do, but Computer Palace has our permission and gives a royalty to the club for it. These sales are to non-members, and we get many members from these buyers.

8. We do not allow advertising and do not sell or give our mailing lists to anyone.

9. We actively trade newsletters and disks with many other clubs, and try to help new ones get started all over the world.

10. Always give credit when credit is due.

11. We accept products for review, do not copy them, and allow the person who reviews one to keep it. We also allow many people to do the reviews, although we favor those who come to "collating parties" to put together the newsletter.

12. We avoid filling the newsletter with editorials, reports of our meetings, or other personal information of no interest to the majority of our members.

13. No pirating!!

Of course the main reason for our success is our members who have always supported us with articles and money when needed. You can continue to do so — please send in articles, reviews, programs, hardware projects or anything you want to share with others — we welcome it. Articles and reviews from children are also very welcome. And if you are a heavy user of the BBS and are able, donations are most appreciated.

For the future, we are considering going to bulk mail, and using the tremendous amount of money we will save to increase the size of the newsletter so as to better cover both the 8 and 16 bit world — but we will need more articles from you to do so.

One of our newest members, teenager Chuck Walbourn (9035 Brickwood, San Antonio, TX 78250) is a programmer and wants to talk to fellow programmers to get and give help. You can also get him a CIS 73537,527 and DELPHI: DIGITA.

In the last issue, it was not completely clear that the two games were from one of my favorite Atari publications, PAGE 6 (18 Underwood Close, Parkside, Stafford, England ST16 1TB). This month we have a series of articles and programs by Rita Plukss and Dick Kellett from another of my favorite newsletters, the Melbourne Australian Atari Gazette (MACE, POB 340, Rosanna, Victoria, Australia 3084) on the fascinating subject of **Mandelbrot Sets**. They were kind enough to send us a disk with their articles and a slide show of the beautiful pictures generated. If you want, you can get the double-sided disk with the programs on one side and the pictures on the other for \$10 from the library. Ask for the Mandelbrot disk — if you have any interest in this subject you will be very impressed. If any of you out there are already doing work in this area, send me your programs and I'll send you the disk free and forward yours to Rita — or you can write her directly at MACE.

Due to several requests, we also have a new disk from the Atari Addicts Club, **Atari Addicts Sector Tool**, a well documented sector editor which works in single or double density and with interesting features. \$5.

Our public domain software utilities and DOS' for the 256K 8-bit Ataris have increased to 2 double sided disks, including a special MACH-DOS. If you want it all, \$20 including the documentation to increase the memory (specify model) or \$10 for the basic software and reprint (up from \$7 — there is too much now included).

If you like text adventure games, **Mysterious Mansion** by Tim Roberson has been improved and expanded — well worth \$5 for a disk full, \$5.

(Minimum price is \$10 a disk, which includes 2 sides full. The new library list is ready at \$1 or free with order-ask for it.)

## VP RAMBLINGS

As we mentioned last month, Mike Dunn is going to COMDEX and should come back with the latest in Atari news, so that next months issue should contain everything fit to print about Atari and maybe even things that arn't. Look for it.

Every month we try to bring you up to date on everything from rumors to new products, but I always try to keep one foot on the ground and tell you that if nothing new comes along for your 8-bit machine there is so much still available to use, and if there wasn't what you have should keep you going for a long time. If you are only a game player then it could be a slow year for you, but if you use your machine for business, word processing etc. then you can go on and on with the 8-bit and never miss using your machine. It's a nice feeling to know that barring a break down in the machine itself there is so much you can do that you should be able to use it forever.

One of the reasons I bring all this up is that we should let all the software houses know we are still out there and they should continue to develop programs for us, and with what they know now about 8-bit programming we should see more sophisticated programs, taking fuller advantage of the Atari's full line of 8-bit machines. While some software houses have done this most have not and they should as there is still a vast market out there which has not really been tapped. Let's try to make them tap it. Write them (software developers) and let them know how we feel. Thanks.

I want to thank all of you who renewed your membership after our call for funds, but remember we have a very extensive library, please use it not only for the good programs you will receive, but it will help us with our cash flow. Thanks again for your support.

— Larry Gold

## LETTER:

I am proud to say I am the A.C.E. representative in Colombia and I feel immensely happy for the Award (Antic Outstanding User Group Award) recently given to ACE. It must be noticed that the award was not given for its being old or full of members but because of the quality that all our directors, present and past, have given to the club through the personal sacrifice, interest and effort in favor of the rest of us Atarians of the club and of the world. Full of emotion and gratitude I give you directors my applause and I consider myself the representative of those lazy members who prefer to be silent in this so important occasion. Here is our standing ovation. With congratulations and respect,

— Fernando Hidalgo  
Cali, Colombia



5



# MANDELBROT SET CON'T

```

410 FOR CHANGE=DLIST+6 TO DLIST+204:IF
  PEEK(CHANGE)=15 THEN POKE CHANGE,14
420 IF PEEK(CHANGE)=79 THEN POKE CHANG
  E,78
430 NEXT CHANGE:POKE 87,7:RETURN
440 REM INITIAL SCREEN
450 ? CHR$(125):? ? "Once design is c
  omplete, press SELECT to save to disk.
  The default name is PICTURE"
460 POKE 764,255:?"Press START to plo
  t a new picture      without saving cur
  rent screen"
470 ? ? "Enter Filename to save desig
  n":?"DO NOT USE DEVICE NAME OR EXTEND
  ER":INPUT B$
480 P=LEN(B$):IF P=0 THEN A$="D:PICTUR
  E":GOTO 510
490 A$(1,3)="D:":FOR X=3 TO P+2:A$(X,X
  )=B$(X-2,X-2):NEXT X
500 A$(P+3,P+6)="PIC"
510 RETURN
520 REM -----
530 REM SAVE TO DISK ROUTINE
540 CLOSE #1:OPEN #1,8,0,A$
550 S=PEEK(88)+256*PEEK(89):TOP=S+7680
  :B$=TOP-S
560 HI=INT(B$/256):LO=B$-(HI*256):POKE
  850,11:POKE 852,PEEK(88):POKE 853,PEE
  K(89):POKE 856,LO:POKE 857,HI
570 D=USR(ADR("hhhh",LO),16):CLOSE #1
580 FOR TIME=1 TO 10:FOR X=10 TO 0 STE
  P -.3:SOUND 0,0,2,X:NEXT X:NEXT TIME
590 GOTO 90

```

## MIRACLES BY BARKLEY

```

10 REM DECORATIVE BORDERS AND PATTERNS 0);
  FOR ATARI WITH EPSON-COMPATIBLE DOT-M
  ATRIX GRAPHICS
15 REM ADAPTED AND EXTENDED BY R.A.BAR
  KLEY FROM J.M. DAVENPORT'S "GRAPHICS F
  OR THE DOT-MATRIX PRINTER"
20 DIM A(16),B(16),E(8),X(8),Y(8),GR$(
  5),BYTES(8),MBYTES(8),A$(3)
22 PRINT CHR$(125)
23 ? ? ? ? ? ? ? ? "TURN
  ON PRINTER."
24 FOR WAIT=1 TO 999:NEXT WAIT
25 LPRINT CHR$(27);"Q";
30 E(0)=1:FOR P=1 TO 7:E(P)=E(P-1)*2:M
  EXT P

```

```

35 REM *** MENU PAGE ***
40 ? CHR$(125):POSITION 2,5
41 ? "DECORATIVE BORDERS AND PATTERN
  5"
42 ? "X<***====+---/\ /\---++====***
  >>>"
43 ? "FOR EPSON-COMPATIBLE PRINTERS
  0"
44 ? ? ?
45 ? ? ? 1 ENTER YOUR OWN PATTERN
  ?"
46 ? ? ? 2 GENERATE RANDOM PATTERN
  N5?"
47 ? ? ? ? " ??:INPUT AN$;IF AN$(1
  OR AN$)2 THEN 90
50 ? "LENGTH OF DOT-PATTERN (TO 16)":I
  NPUT Q:IF Q<0 OR Q>16 THEN 90
55 ? "PRINTER DENSITY CODE (0 TO 6)":I
  NPUT D:IF D<0 OR D>6 THEN 90
60 ? "NO. OF REPETITIONS (0= ONE LINE)
  ":INPUT R:IF R<0 THEN 90
65 IF AN$=2 THEN ? "NO. OF DIFFERENT P
  ATTERNS TO PRINT":INPUT N:GOSUB 610:GO
  TO 100
70 IF AN$=1 THEN N=1
75 GOSUB 510:GOTO 100
80 END
90 ? CHR$(125):POSITION 0,5:GOTO 45
100 FOR PATTERN=1 TO N
130 C=0
140 G2=INT((40*Q)/256):G1=(40*Q-256*G2
  )+1
150 GR$(1,1)=CHR$(27):GR$(2,2)="*":GR$(
  3,3)=CHR$(0):GR$(4,4)=CHR$(G1):GR$(5,
  5)=CHR$(G2)
160 LPRINT GR$;
170 FOR J=1 TO 20
180 FOR I=1 TO Q:LPRINT CHR$(A(I));:NE
  XT I
190 FOR I=Q TO 1 STEP -1:LPRINT CHR$(A
  (I));:NEXT I
200 NEXT J
210 LPRINT CHR$(27);"A";CHR$(8);CHR$(1
  78
220 LPRINT GR$;
230 FOR J=1 TO 20
240 FOR I=1 TO Q:LPRINT CHR$(B(I));:NE
  XT I
250 FOR I=Q TO 1 STEP -1:LPRINT CHR$(B
  (I));:NEXT I
260 NEXT J
270 C=C+1:IF C<R THEN LPRINT CHR$(27)
  ;"A";CHR$(8);CHR$(13):GOTO 160
280 LPRINT CHR$(27);"2";CHR$(13);
  T I
290 FOR I=1 TO Q:LPRINT A(I);" ";:NEX
  T I
300 LPRINT CHR$(27);"2";CHR$(13);
310 FOR I=1 TO Q:LPRINT B(I);" ";:NEX
  T I

```

```

312 LPRINT CHR$(27);"2";CHR$(13);
315 LPRINT " PATTERN # ";PATTERN;"
  DOT PATTERN LENGTH: ";Q;" PRINTE
  R DENSITY CODE: ";D
320 LPRINT CHR$(27);"2";CHR$(13);
325 IF N)1 AND PATTERN<N THEN GOSUB 61
  0
330 NEXT PATTERN
340 ? ? ? PRESS <RETURN>...:
  INPUT A$:IF A$="" THEN CONT
350 ? CHR$(125):GOTO 45
500 REM *** ENTER OWN PATTERN NO.5 ***
510 ? ? ? ENTER ";Q;" NUMBERS ( 0 -
  255)"
520 ? ? ? No. Pin Pattern"
522 ? ? ? 7 0":?
530 FOR I=1 TO Q:?" CHR$(28);
540 INPUT A$:A(I)=VAL(A$):L=LEN(A$)
550 GOTO 700
600 REM ** ENTER RANDOM SEQUENCE **
610 ? ? ? No. Pin Pattern"
612 ? ? ? 7 0":?
615 REM FOR K=1 TO N
620 FOR I=1 TO Q:A(I)=INT(RND(0)*(255)
  )
630 A$=STR$(A(I)):L=LEN(A$)
700 REM ** RIGHT JUSTIFY A(I) **
710 ? CHR$(28);" ";:IF L=3 THEN ?
  A(I);:GOTO 740
720 IF L=2 THEN ? " ";A(I);:GOTO 740
730 IF L=1 THEN ? " ";A(I);
740 ? " - ";
1000 REM ***MIRROR-IMAGE CONVERSION***
1010 REM * DIGITAL TO BINARY *
1020 CODE=A(I)
1030 FOR P=7 TO 0 STEP -1
1040 K(P)=CODE-E(P)
1050 IF K(P)<0 THEN Y(P)=0:GOTO 1070
1060 Y(P)=1:CODE=K(P)
1070 IF Y(P) THEN ? CHR$(148);:GOTO 10
  78
1075 ? CHR$(160);
1078 NEXT P
1080 ? ? ? FOR J=0 TO 7
1082 REM * REVERSE BIT SEQUENCE *
1085 BYTES(J+1,J+1)=CHR$(Y(J))
1090 MBYTES(8-J,8-J)=CHR$(Y(J))
1095 NEXT J
1100 REM * BINARY TO DIGITAL *
1110 B(I)=128*Y(0)+64*Y(1)+32*Y(2)+16*
  Y(3)+8*Y(4)+4*Y(5)+2*Y(6)+Y(7)
1200 NEXT I
1210 ? ? ? CODE MIRROR IMAGE:"
1220 FOR I=1 TO Q:PRINT " ";A(I);,B(
  I):NEXT I:?"
1500 RETURN

```



# GUESS!!!

```

10 GOSUB 10000
15 POSITION 0,20: ? #6;"
";
20 POSITION 0,20: ON P+2 GOTO 40
30 ? #6;"PLAYER & 1": GOTO 50
40 ? #6;"player / -"
50 POSITION 0,22: ? #6;"ACROSS 2 ";
60 G=- (P-1)*32
70 GET #1,Z: Y=Z-48: IF Y<1 OR Y>6 THEN
70
80 ? #6;CHR$(128+Z): ? #6;"DOWN 2 ";
K=(Y-1)*3
90 GET #1,Z: Y=Z-48: IF Y<1 OR Y>6 THEN
50
100 ? #6;CHR$(128+Z): Y=(Y-1)*3
110 IF SGN(A(X,Y))=-P THEN 15
120 GOSUB 1000: GOSUB 9000
130 IF ABS(A(X,Y))<C THEN P=-P: GOTO 15

140 GOSUB 2000
150 I1=0: I2=0: Z=0: FOR K=0 TO 18 STEP 3
: FOR Y=0 TO 18 STEP 3
160 IF A(X,Y)<0 THEN I1=1
170 IF A(X,Y)>0 THEN I2=1
180 GOSUB 9000
190 IF ABS(A(X,Y))>C THEN Z=1: GOSUB 1
010: GOSUB 2000
200 NEXT Y: NEXT K
210 IF I1=0 THEN POSITION 0,20: ? #6;"P
LAYER & 1 ": ? #6;"WINS": GOTO 500
220 IF I2=0 THEN POSITION 0,20: ? #6;"P
layer / - ": ? #6;"wins": GOTO 500
230 IF Z=1 THEN 150
240 P=-P: GOTO 15
500 ? #6;"another game
"; GET #1,K: IF K=89 THEN RUN
510 ? : END
1000 A(X,Y)=A(X,Y)+P
1010 POSITION X+1,Y+1: ? #6;CHR$(G+39);
CHR$(G+39); CHR$(G+39);
1015 F=ABS(A(X,Y))
1020 POSITION X+1,Y+2: ? #6;CHR$(G+39);
CHR$(F+48+G); CHR$(G+39);
1030 POSITION X+1,Y+3: ? #6;CHR$(G+39);
CHR$(G+39); CHR$(G+39);
1040 RETURN
2000 POSITION X+2,Y+1: ? #6;CHR$(G+35)
2010 POSITION X+1,Y+2: ? #6;CHR$(G+37)
2020 POSITION X+3,Y+2: ? #6;CHR$(G+38)
2030 POSITION X+2,Y+3: ? #6;CHR$(G+36)
2040 FOR J=15 TO 0 STEP -0.15: SOUND 0,
89,0,J: NEXT J
2050 A(X,Y)=A(X,Y)-C*P: IF A(X,Y)<0 TH
EN GOSUB 1010: GOTO 2140

```

```

2060 IF (X/2=INT(X/2) AND Y/2=INT(Y/2)
) OR (X/2<INT(X/2) AND Y/2<INT(Y/2))
THEN 2110
2070 POSITION X+1,Y+1: ? #6;" "
2080 POSITION X+1,Y+2: ? #6;" "
2090 POSITION X+1,Y+3: ? #6;" "
2100 GOTO 2140
2110 POSITION X+1,Y+1: ? #6;" "
2120 POSITION X+1,Y+2: ? #6;" "
2130 POSITION X+1,Y+3: ? #6;" "
2140 A=X: B=Y: M=P
2150 IF A<3 THEN 2180
2160 A(A-3,B)=ABS(A(A-3,B))*M+M
2170 X=A-3: Y=B: GOSUB 1010
2180 IF A>13 THEN 2210
2190 A(A+3,B)=ABS(A(A+3,B))*M+M
2200 X=A+3: Y=B: GOSUB 1010
2210 IF B<3 THEN 2240
2220 A(A,B-3)=ABS(A(A,B-3))*M+M
2230 X=A: Y=B-3: GOSUB 1010
2240 IF B>13 THEN 2270
2250 A(A,B+3)=ABS(A(A,B+3))*M+M
2260 X=A: Y=B+3: GOSUB 1010
2270 RETURN
9000 C=4
9010 IF X=15 OR X=0 THEN C=C-1
9020 IF Y=15 OR Y=0 THEN C=C-1
9030 RETURN
10000 GRAPHICS 0: POKE 752,1: POKE 709,1
5: POKE 710,24: POSITION 8,1: POKE 82,1
10010 ? "EXPLOSION BY P. GIBBS": POSITI
ON 8,2: ? "
10020 ? "4 Explosion is a two player
game.The"
10030 ? "object is to leave your oppon
ent with"
10040 ? "no pieces on the board."
10050 ? "4 In a go you may place a p
iece in"
10060 ? "either an unoccupied square,
or one"
10070 ? "occupied by your own pieces."
10080 ? "4 There is a maximum number
of pieces"
10090 ? "that can remain in any square
In most"
10100 ? "squares this number is three
out at the";
10110 ? "edges it is two and at the co
rners it"
10120 ? "is one. If that number is ext
eded the"
10130 ? "pieces on that square will ex
tode into";

```

```

10140 ? "those surrounding it, so taki
ng them over."
10150 ? "4 Multiple explosions are p
ossible !4"
11000 FOR I=0 TO 512: POKE 25600+I, PEEK
(57344+I): NEXT I
11010 FOR I=8 TO 63: READ A: POKE 25600+
I, 255-A: NEXT I
11020 FOR I=136 TO 183: POKE 25600+I, 25
5-PEEK(57344+I): NEXT I
12000 ? " PRESS ANY KEY TO PLA
Y ";
12910 DIM A(20,20): FOR I=0 TO 18 STEP
3: FOR II=0 TO 18 STEP 3: A(I,II)=0: NEXT
II: NEXT I
12020 OPEN #1,4,0,"K": GET #1,K
12500 GRAPHICS 17: POKE 708,54: POKE 756
,100: POKE 710,28: POKE 712,0
13000 ? #6;" 1 2 3 4 5 6"
13010 FOR I=0 TO 17 STEP 3
13020 ON (I/2=INT(I/2)) GOTO 13040
13030 FOR L=0 TO 2: ? #6;" "
13040 FOR L=0 TO 2: ? #6;" "
13050 FOR I=0 TO 15 STEP 3
13060 POSITION 0,2+I: ? #6;CHR$(177+(I/
3)): NEXT I
14000 P=1: RETURN
20000 DATA 170,170,170,170,170,170,170
,170
20010 DATA 255,0,255,0,255,0,255,0
20020 DATA 24,60,126,219,24,24,24,24
20030 DATA 24,24,24,24,219,126,60,24
20040 DATA 16,48,96,255,255,96,48,16
20050 DATA 8,12,6,255,255,6,12,8
20060 DATA 0,0,0,0,0,0,0,0
32000 GOSUB 12500: GOTO 20
32500 GOTO 32500

```

## ACTION

```

BYTE FUNC EQ$(CARD $1,$2)
BYTE TEMP
;
; RETURNS TRUE($FF) IF STRINGS ARE
; EQUAL, FALSE($00) IF STRINGS ARE
; NOT EQUAL
; $1 IS ADDRESS OF FIRST STRING
; $2 IS ADDRESS OF SECOND STRING
IF SCOMPARE($1,$2) THEN
TEMP=$00
ELSE
TEMP=$FF
FI
RETURN (TEMP)

```



# OCKERS IN ACTION

```
*****
;* Demo using counters provided by *
;* the program itself, decremented *
;* during the Vertical Blank Int. *
;*****
```

```
BYTE ARRAY cntdn(5),cntflg(5),maxcnt(5)
]
BYTE ARRAY rot=[47 124 92 45 45]
```

```
PROC Setvect=$E45C(BYTE areg,xreg,yreg)
]
; This time used to insert routine into
; vertical blank interrupt
```

```
PROC Vblank() ; Routine to be inserted
```

```
BYTE i
FOR i=0 TO 4 DO
  IF cntdn(i)>0 THEN
    cntdn(i)--1 FI ;decrement counters
  IF cntdn(i)=0 THEN
    cntflg(i)=0 FI ;set flag if zero
  OD
  [54C 55F 5E4] ; In line code jump back
```

```
PROC Rotate() ;Update figures at timeout
```

```
BYTE j
FOR j=0 TO 4 DO
  WHILE maxcnt(j)<15 AND cntflg(j)=0
```

```
DO
  IF rot(j)=47 THEN rot(j)=45
  ELSEIF rot(j)=45 THEN rot(j)=9
```

```
2 ELSEIF rot(j)=92 THEN rot(j)=1
```

```
24 ELSEIF rot(j)=124 THEN rot(j)=
```

```
47 FI
Position(5*j+10,10) Put(rot(j))
```

```
] IF Rand(15)<1 THEN maxcnt(j)=
```

```
+1 FI
cntdn(j)=maxcnt(j) cntflg(j)=1
```

```
OD
```

```
OD
```

```
RETURN
```

```
PROC Spin5()
```

```
BYTE k
```

```
Put(125) Poke(752,1) ;Clear screen
Setvect(6,Vblank RSH 8,Vblank & $FF)
```

```
;Our vertical blank routine inserted
```

```
DO ;do forever
  IF Rand(0)<1 AND RAND(30)<1 THEN
```

```
k=Rand(5) maxcnt(k)=1 cntflg(k)=
```

```
1 FI ;Rotations re-started at random
```

```
M Rotate() ;Update all
```

```
OD
RETURN
```

```
*****
```

```
; A Demonstration of using system
```

```
; timer #3, watching a flag
```

```
; location for timeout.
```

```
*****
```

```
PROC Setime=58460(BYTE areg,xreg,yreg)
```

```
; Used to set timers during VBI
```

```
PROC Pollflag() ;Poll flag for timeout
```



NOT ACTION BY  
OCKERS



# FIXACEC IN C

```

/*=====*/
/* File: FIXACEC.C */
/*=====*/
/* Fix "ftoi()" in ACE-C */
/*
/* ACE Newsletter
/* 3662 Vine Maple, Eugene, Or
/* 97405 $14 yr May '86
/*
/* By A.J.Aspromatis
/*
/* Org: 3/21/86 Rev: 4/ 7/86 */
/*=====*/
/**/
#define bufr 0x5000
#define bug_loc 0x950
/**/
main()
$(
    char data;
    int count,count2,line,x;
    /* setup */
    poke(752,1); /* cursor off */
    putchar('\f'); /* clear screen */
    position(10,2);
    printf("ftoi() FIX FOR ACEC");
    position(11,3);
    printf("by A.J.Aspromatis");
    line=10;
    position(3,line);
    printf("Insert ACEC disk & press <RE
TURN>");
    while(getkey()!='\n');
    /* open file to read */
    open_file(4,"read",line++);
    position(4,++line);
    printf("Reading ENGINE.OBJ");
    /* get file data */
    for(count=0; ++count)
    $(
        if((data=cgetc(1))<0) break; /* re
ad until EOF */
        poke(bufr+count,data);
    $)
    close(1);
    /* test if fixed */
    if(dpeek(bufr+bug_loc)==0xD4A6 && dp
eek(bufr+(bug_loc)+2)==0xD5A5)
    $(
        /* fix it */
        poke(bufr+bug_loc,0xA5);
        poke(bufr+(bug_loc)+2,0xA6);
        position(4,++line);
        printf("---fixed---");
        /* open file to write */
        open_file(8,"write",line);

```

```

        position(4,++line);
        printf("Writing ENGINE.OBJ");
        /* put file data */
        for(count2=0; count2<count; ++count2
) cputc(peek(bufr+count2),1);
        close(1);
    $)
    else
    /* do not fix */
    $(
        position(3,++line);
        if(dpeek(bufr+bug_loc)==0xD4A5 &&
dpeek(bufr+(bug_loc)+2)==0xD5A6)
            printf("File already fixed");
        else printf("Different version of
ENGINE.OBJ");
        position(4,++line);
        printf("Fix aborted");
    $)
    position(3,20);
    printf("Done -- press <RETURN> for D
OS");
    while(getkey()!='\n');
    $)
/**/
/*=====*/
/* Function definitions */
/*=====*/
/**/
/*-----*/
/* ERROR CODE NOTICE */
notice(error,line)
char error;
int line;
$(
    error=-error;
    position(3,++line);
    printf("Error %d on disk access.",e
rror);
    position(3,++line);
    printf("Press <RETURN> to try again
.");
    while(getkey()!='\n');
    line=erase(2,line);
    return;
$)
/**/
/*-----*/
/* ERASE LINES */
erase(number,line)
int line,number;
$(
    for(; number!=0; --number, --line)
    $(
        position(2,line);
        printf("

```

```

");
    $)
    return(line);
$)
/**/
/*-----*/
/* OPEN FILE TO READ/WRITE */
open_file(cmd,prompt,line)
char prompt[];
int cmd,line;
$(
    char code;
    while((code=open(1,cmd,-1,"D:ENGINE
.OBJ"))<0)
    $(
        position(4,++line);
        printf("Cannot %s ENGINE.OBJ",prom
pt);
        notice(code,line);
        close(1);
        line=erase(1,line);
    $)
    return;
$)

```

## ACTION ACTION

```

BYTE FUNC MES(CARD S1,S2)
BYTE TEMP
;
; RETURNS FALSE($00) IF STRINGS ARE
; EQUAL, TRUE($FF) IF STRINGS ARE
; NOT EQUAL
;
; S1 IS ADDRESS OF FIRST STRING
; S2 IS ADDRESS OF SECOND STRING
;
IF SCOMPARE(S1,S2) THEN
    TEMP=$FF
ELSE
    TEMP=$00
FI
RETURN (TEMP)

```

PROC GLITCH()

[2]

RETURN



# MANDELBROT SET

```
10 REM MANDELZOOM PART 1 - MANDLAL
20 REM refer to Scientific American
    August 1985
30 REM MACE subset of Mandelbrot set
    by DICK KELLET
```

```
40 REM MACE MARCH 1986
50 REM *****
60 REM *****
70 SETCOLOR 2,0,0: CHR$(125); " WHEN
  THE PROGRAM IS RUN THE SCREEN WILL BL
  ANK TO SPEED THE CALCULATIONS":?
80 ? " EVEN WITH THIS THE TOTAL TIM
  E TO CALCULATE AND SAVE THE DATA IS B
  ETWEEN 10 AND 20 HOURS":?
90 ? " WHILE THE PROGRAM IS RUNNING
  PRESS ANY FUNCTION KEY (OPTION, SELECT
  OR START) TO DISPLAY THE ";
100 ? "CURRENT NUMBER OF ITERATIONS,
  HORIZONTAL COUNT AND VERTICAL COUNT.
  "
110 ? :? " PRESS START TO CONTINU
  E"
```

```
120 IF PEEK(53279)<7 THEN GOSUB 490
130 ? CHR$(125):DIM MANS(8),FILES(15)
140 ? "INSERT DISK WITH AT LEAST 250 F
  REE SECTORS":?
150 ? "INPUT FILE NAME. DO NOT USE DEV
  ICE OR EXTENDER. ";
160 ? "THE PROGRAM WILL SUPPLY A DEV
  ICE NAME D. AND A .DAT EXTENDER"
170 ? "ENTER NAME OF DATA FILE":? "(MA
  X 8 CHARACTERS)":INPUT MANS$
180 FILES(1,2)="D:";FILES(3)=MANS$:FILE
  $(LEN(MANS$)+3,LEN(MANS$)+7)=".DAT"
190 CLOSE #1:OPEN #1,8,0,FILES$
200 HS=159:VS=191:IT=100
210 TRAP 210: CHR$(125); "ACORNER ,BCO
  RNER ARE THE BOTTOM LEFT CORNER OF TH
  E AREA TO BE PLOTTED"
220 ? "ENTER ACORNER,BCORNER ":INPUT
  ACNR,BCNR
230 TRAP 230:POSITION 14,10:?" "
```

```
240 POSITION 2,5:?" SIDE IS THE DISTA
  NCE TO BE COVERED IN A POSITIVE GOING H
  ORIZONTAL DIRECTION"
250 ? " THE DISTANCE COVERED IN A PO
  SITIVE GOING VERTICAL DIRECTION IS APP
  ROX. 0.6 THE HORIZONTAL DISTANCE"
260 ? "ENTER SIDE ":INPUT SD
270 SCR=PEEK(559):POKE 559,0:POKE 755,
  0: CHR$(125)
280 GAP=SD/HS:VGAP=GAP/2:REM ADJUST FO
  R PIXEL SHAPE
290 REM MAIN NUMBER CRUNCHING ROUTINES
```

```
300 FOR V=0 TO VS:FOR H=0 TO HS
310 BC=V*VGAP+BCNR
320 AC=H*GAP+ACNR
330 AZ=0:BZ=0:I=0:J=0:K=0:COUNT=0
340 AZ=AC+I-J:BZ=BC+2*K
350 COUNT=COUNT+1
360 I=AZ*AZ:J=BZ*BZ:K=AZ*BZ
370 REM DISPLAY CURRENT STATE WHEN A F
  UNCTION KEY IS PRESSED
380 IF PEEK(53279)<7 THEN GOSUB 490
390 REM BLANK SCREEN WHEN FUNCTION KEY
  IS RELEASED TO SPEEDUP CALCULATIONS
400 IF PEEK(53279)=7 THEN POKE 559,0
410 IF COUNT<IT AND (I+J)<4 THEN 340
420 PUT #1,COUNT
430 COUNT=0:NEXT H:NEXT V
440 PUT #1,155:REM EOL
450 CLOSE #1:POKE 559,SCR:GRAPHICS 18:
  POSITION 5,2:?"#6:"FINISHED":POSITION
  3,4:?"#6:"FILE SAVED OK"
460 POSITION 3,6:?"#6:FILES:POKE 755,2
  470 GOTO 470
480 REM DISPLAY CURRENT STATE
490 POKE 77,0:POKE 559,SCR:POSITION 2,
  2:?"COUNT"
500 POSITION 15,2:?"HOR. POS.":POSITI
  ON 30,2:?"VERT. POS."
510 POSITION 0,3:?"-----+-----"
  |-----+-----|
  |-----+-----|
  |-----+-----|
520 POSITION 4,4:?"COUNT; " ":POSITION
  18,4:?"H; " ":POSITION 34,4:?"V; " "
530 RETURN
```

```
10 REM MANDELZOOM - PART 2 MANDPLOT
20 REM refer to Scientific American
    August 1985
30 REM This reads the data file from
  MANDLAL and plots the screen
40 REM
50 REM MACE subset of Mandelbrot set
  by DICK KELLET
60 REM MACE - MARCH 1986
70 REM *****
80 REM *****
90 CLR :DIM A$(15),B$(15),LD$(15),FILE
```

```
$(8):GOSUB 450
100 ? CHR$(125):?"ENTER FILE NAME OF
  DATA FILE.DO NOT USE DEVICE NAME OR
  EXT.":INPUT FILES$
110 LD$(1,3)="D:";LD$(3)=FILES$:LD$(LEN
  (FILES$)+3)=".DAT"
120 TRAP 120:?"CHR$(125)
130 ? "ENTER [ ] TO DISPLAY PICTURE WITH
  USER SELECTED COUNTS FOR COLOR CHANG
  E"
140 ? "ENTER [ ] TO DISPLAY PICTURE WITH
  COLOR CHANGE MOD 3"
150 ? "ENTER [ ] TO PLOT THE COUNTS LOWE
  R THAN THE FIRST CHANGE POINT IN MOD 3
  "
160 INPUT CH:IF CH<1 OR CH>3 THEN 120
170 TRAP 40000:IF CH=2 THEN 250
180 ? CHR$(125):?"THE COUNTS FOR CHAN
  GE ARE BETWEEN 1 AND 100. ENTER THE
  CHANGE NUMBERS IN ASCENDING ORDER"
190 TRAP 190:POSITION 3,5:?" "
200 POSITION 3,4:?"ENTER COUNT FOR FI
  RST COLOR CHANGE ":INPUT C1:IF C1<1 OR
  C1>100 THEN 190
210 TRAP 210:POSITION 3,7:?" "
220 POSITION 3,6:?"ENTER COUNT FOR SE
  COND COLOR CHANGE ":INPUT C2:IF C2<=C1
  OR C2>100 THEN 210
230 TRAP 230:POSITION 3,9:?" "
240 POSITION 3,8:?"ENTER COUNT FOR TH
  IRD COLOR CHANGE ":INPUT C3:IF C3<=C2
  OR C3>100 THEN 230
250 TRAP 350
260 GOSUB 400
270 CLOSE #1:OPEN #1,4,0,LD$
280 FOR V=191 TO 0 STEP -1:FOR H=0 TO
  159:GET #1,N:COL=N
290 IF CH=1 OR CH=3 THEN COL=1*(N<C1)+
  3*(N<C2) AND (N<C3)+2*(N<C2) AND
  (N<C3)+1*(N<C3)
300 IF CH=2 OR (CH=3 AND (N<C1)) THEN
  COL=(N/3-INT(N/3))*3+1
310 IF N=100 THEN COL=0
320 COLOR COL:IF V<96 THEN POKE 89,SA-
  15:PLOT H,V:REM TOP HALF OF SCREEN
330 IF V>95 THEN POKE 89,SA:PLOT H,V-9
  6:REM BOTTOM HALF OF SCREEN
340 NEXT H:NEXT V
350 POKE 89,SA-15:CLOSE #1:REM RETURN
  POINTER TO TOP HALF OF SCREEN
360 IF PEEK(53279)=6 THEN 90
370 IF PEEK(53279)=5 THEN 540
380 GOTO 360
390 REM SETUP GRAPHICS 7+ SCREEN
400 GRAPHICS 8+16:DLIST=PEEK(560)+PEEK
  (561)*256:POKE DLIST+3,78:SA=PEEK(89)+
  15
```



# EXPLOSION

```

10 GOSUB 10000
15 POSITION 0,20: ? #6;"
";
20 POSITION 0,20:ON P+2 GOTO 40
30 ? #6;"PLAYER & 1":GOTO 50
40 ? #6;"player / -"
50 POSITION 0,22: ? #6;"ACROSS 2 ";
60 G--(P--1)*32
70 GET #1,Z:Y=Z-48:IF Y<1 OR Y>6 THEN
70
80 ? #6;CHR$(128+Z): ? #6;"DOWN 2 ";:
K=(Y-1)*3
90 GET #1,Z:Y=Z-48:IF Y<1 OR Y>6 THEN
50
100 ? #6;CHR$(128+Z):Y=(Y-1)*3
110 IF SGN(A(X,Y))=-P THEN 15
120 GOSUB 1000:GOSUB 9000
130 IF ABS(A(X,Y))<C THEN P=-P:GOTO 15
140 GOSUB 2000
150 I1=0:I2=0:Z=0:FOR K=0 TO 18 STEP 3
:FOR Y=0 TO 18 STEP 3
160 IF A(X,Y)<0 THEN I1=1
170 IF A(X,Y)>0 THEN I2=1
180 GOSUB 9000
190 IF ABS(A(X,Y))=C THEN Z=1:GOSUB 1
010:GOSUB 2000
200 NEXT Y:NEXT K
210 IF I1=0 THEN POSITION 0,20: ? #6;"P
LAYER & 1 ": ? #6;"WINS":GOTO 500
220 IF I2=0 THEN POSITION 0,20: ? #6;"p
layer / - ": ? #6;"wins":GOTO 500
230 IF Z=1 THEN 150
240 P=-P:GOTO 15
500 ? #6;"Another game
":GET #1,K:IF K=89 THEN RUN
510 ? :END
1000 A(X,Y)=A(X,Y)+P
1010 POSITION X+1,Y+1: ? #6;CHR$(G+39);
CHR$(G+39);CHR$(G+39);
1015 F=ABS(A(X,Y))
1020 POSITION X+1,Y+2: ? #6;CHR$(G+39);
CHR$(F+48+G);CHR$(G+39);
1030 POSITION X+1,Y+3: ? #6;CHR$(G+39);
CHR$(G+39);CHR$(G+39);
1040 RETURN
2000 POSITION X+2,Y+1: ? #6;CHR$(G+35)
2010 POSITION X+1,Y+2: ? #6;CHR$(G+37)
2020 POSITION X+3,Y+2: ? #6;CHR$(G+38)
2030 POSITION X+2,Y+3: ? #6;CHR$(G+36)
2040 FOR J=15 TO 0 STEP -0.15:SOUND 0,
89,8,J:NEXT J
2050 A(X,Y)=A(X,Y)-C*P:IF A(X,Y)<0 TH
EN GOSUB 1010:GOTO 2140
2060 IF (X/2=INT(X/2) AND Y/2=INT(Y/2))

```

```

) OR (X/2<INT(X/2) AND Y/2<INT(Y/2))
THEN 2110
2070 POSITION X+1,Y+1: ? #6;"
2080 POSITION X+1,Y+2: ? #6;"
2090 POSITION X+1,Y+3: ? #6;"
2100 GOTO 2140
2110 POSITION X+1,Y+1: ? #6;"
2120 POSITION X+1,Y+2: ? #6;"
2130 POSITION X+1,Y+3: ? #6;"
2140 A=X:B=Y:N=P
2150 IF A<3 THEN 2180
2160 A(A-3,B)=ABS(A(A-3,B))*N+M
2170 X=A-3:Y=B:GOSUB 1010
2180 IF A>13 THEN 2210
2190 A(A+3,B)=ABS(A(A+3,B))*N+M
2200 X=A+3:Y=B:GOSUB 1010
2210 IF B<3 THEN 2240
2220 A(A,B-3)=ABS(A(A,B-3))*N+M
2230 X=A:Y=B-3:GOSUB 1010
2240 IF B>13 THEN 2270
2250 A(A,B+3)=ABS(A(A,B+3))*N+M
2260 X=A:Y=B+3:GOSUB 1010
2270 RETURN
9000 C=4
9010 IF X=15 OR X=0 THEN C=C-1
9020 IF Y=15 OR Y=0 THEN C=C-1
9030 RETURN
10000 GRAPHICS 0:POKE 752,1:POKE 709,1
5:POKE 710,24:POSITION 0,1:POKE 82,1
10010 ? "EXPLOSION BY P. GIBBS":POSITI
ON 0,2: ? "
10020 ? "4 Explosion is a two player
game.The"
10030 ? "Object is to leave your oppon
ent with"
10040 ? "no pieces on the board."
10050 ? "4 In a go you may place a p
iece in"
10060 ? "either an unoccupied square,
or one"
10070 ? "occupied by your own pieces."
10080 ? "4 There is a maximum number
of pieces"
10090 ? "that can remain in any square
. In most"
10100 ? "squares this number is three
but at the";
10110 ? "edges it is two and at the co
rners it"
10120 ? "is one. If that number is exc
eeded the"
10130 ? "pieces on that square will ex
plode into";
10140 ? "those surrounding it, so taki
ng them over."

```

```

10150 ? "4 Multiple explosions are p
ossible !4"
11000 FOR I=0 TO 512:POKE 25600+I,PEEK
(57344+I):NEXT I
11010 FOR I=0 TO 63:READ A:POKE 25600+
I,255-A:NEXT I
11020 FOR I=136 TO 183:POKE 25600+I,25
5-PEEK(57344+I):NEXT I
12000 ? " PRESS ANY KEY TO PLA
Y ";
12010 DIM A(20,20):FOR I=0 TO 18 STEP
3:FOR II=0 TO 18 STEP 3:A(I,II)=0:NEXT
II:NEXT I
12020 OPEN #1,4,0,"K":GET #1,K
12500 GRAPHICS 17:POKE 708,54:POKE 756
,100:POKE 710,28:POKE 712,0
13000 ? #6;" 1 2 3 4 5 6"
13010 FOR I=0 TO 17 STEP 3
13020 ON (I/2=INT(I/2)) GOTO 13040
13030 FOR L=0 TO 2: ? #6;"
13040 FOR L=0 TO 2: ? #6;"
13050 FOR I=0 TO 15 STEP 3
13060 POSITION 0,2+I: ? #6;CHR$(177+(I/
3)):NEXT I
14000 P=1:RETURN
20000 DATA 170,170,170,170,170,170,170
,170
20010 DATA 255,0,255,0,255,0,255,0
20020 DATA 24,60,126,219,24,24,24,24
20030 DATA 24,24,24,24,219,126,60,24
20040 DATA 16,48,96,255,255,96,48,16
20050 DATA 8,12,6,255,255,6,12,8
20060 DATA 0,0,0,0,0,0,0,0
32000 GOSUB 12500:GOTO 20
32500 GOTO 32500

```

## ACTION ACTION

```

BYTE FUNC GTS(CARD S1,S2)
BYTE TEMP
;
; RETURNS TRUE($FF) IF S1 <= S2
; FALSE($00) OTHERWISE
;
; S1 IS ADDRESS OF FIRST STRING
; S2 IS ADDRESS OF SECOND STRING
;
IF SCOMPARE(S1,S2)>0 THEN
TEMP=$FF
ELSE
TEMP=$00
FI
RETURN (TEMP)

```



# EXPLOSION CON'T

# NO.0019X1

# SNARK

```

30005 PUL$="P":PUL$(39)="P":PUL$(2)=PU
L$:LFT$="P":LFT$(23)="P":LFT$(2)=LFT$
OPEN #3,4,0,"K":BLK$="K"
30007 PRS$=" " PRESS ANY KEY TO CONTINU
E"
30010 POKE 106,PEEK(740)-6:GRAPHICS 0:
POKE 708,102:POKE 709,44:POKE 710,50:P
OKE 712,0:POKE 752,1: " "WAIT!"
30015 RESTORE 30050:FOR I=1 TO 16:READ
A:CMPS(I,I)=CHR$(A):NEXT I
30020 POKE 512,128:POKE 513,6:FOR I=16
64 TO 1731:READ A:POKE I,A:NEXT I
30025 DLS=PEEK(560)+256*PEEK(561):POKE
DLS+7,7:POKE DLS+8,6:POKE DLS+10,128:
POKE DLS+22,128
30030 X=USR(1700):POKE 54286,192:RETUR
N
30049 REM COMPARISON BYTES - ATARI/XL
FIX
30050 DATA 208,212,96,19,28,47,17,112
30060 DATA 157,69,3,169,0,157,75,3
30069 REM FIRST INTERRUPT ROUTINE
STARTS -1664-
30070 DATA 72,169,112,141,10,212,141,2
4,208,169,156,141,0,2,104,64
30079 REM I/O ROUTINE IOCB
STARTS -1680-
30080 DATA 104,104,104,170,76,86,228
30089 REM I/O ROUTINE RDH
STARTS -1687-
30090 DATA 104,32,83,228,96
30109 REM SECOND INTERRUPT ROUTINE
STARTS -1692- WAS -1714-
30110 DATA 72,169,2,141,10,212,141,24,
208,169,156,141,9,212,104,64
30119 REM VBI SET RTN STARTS -1708-
WAS 1730
30120 DATA 104,160,183,162,6,169,7,32,
92,228,96
30129 REM VBI ROUTINE STARTS -1719-
WAS 1741
30130 DATA 169,128,141,0,2,169,192,141
,14,212,76,98,228

```

## ACTION

```

BYTE FUNC INKEY()
;
; IF A KEY IS PRESSED RETURNS ITS
; ASCII VALUE, 255 OTHERWISE
;
BYTE TEMP
IF PEEK(764)<255 THEN
TEMP=GETD(7)
RETURN (TEMP)
FI
RETURN (255)

```

```

0 REM *****
1 REM -SNARK HUNTER-
2 REM *
3 REM * by:JIM DANKO 10/84 *
4 REM * S.P.A.C.E St.Paul *
5 REM * ATARI Computer Enthusiasts*
6 REM *****
7 FOR A=256 TO 288:READ B:POKE A,B:NEX
T A
8 DATA 104,169,17,141,40,2,169,1,141,4
1,2,169,30,141,26,2,96,173,243,2,41,23
,73,2,141,243,2,169,30,141,26,2,96
9 POKE 284,0:A=USR(256)
10 GRAPHICS 1:POKE 82,2:POKE 752,0: "
? :? :?
12 POSITION 4,15: " #6;"snark"
13 POSITION 10,15: " #6;"hunter"
14 POSITION 14,3: " #6;"n"
15 POSITION 13,4: " #6;"w"
16 POSITION 14,5: " #6;"s"
17 POSITION 15,4: " #6;"e"
18 POSITION 14,4: " #6;"+"
19 POSITION 0,10: " #6;"y":POSITION 10,
0: " #6;"x":GOSUB 9000:CLR
20 FOR A=0 TO 9
30 POSITION A,0: " #6:B
31 B=B+1
40 NEXT A
50 FOR C=0 TO 9
60 POSITION 0,C: " #6:D
61 D=D+1
62 NEXT C
65 REM PLOTTED POINTS
70 FOR A=1 TO 9
80 FOR B=1 TO 9
85 SOUND 0,A*B,10,8:SETCOLOR 3,0,A*B+6
90 POSITION B,A: " #6;" "
100 NEXT B
110 NEXT A
111 SOUND 0,0,0,0
112 OPEN #1,4,0,"K:"
115 REM *****
120 X1=INT(9*RND(1))+1
130 Y1=INT(9*RND(1))+1
135 FOR T=9 TO 0 STEP -1
140 " "INPUT X COORDINATES(1+1): " "
";
150 GET #1,X: " CHR$(X):POKE 53279,4:PO
KE 53279,6
160 " " "INPUT Y COORDINATES(1+1): " "
";
170 GET #1,Y: " CHR$(Y):POKE 53279,4:PO
KE 53279,6
172 IF X<48 OR X>57 OR Y<49 OR Y>57 TH
EN " "REENTER " "GOTO 140
174 X=X-48:Y=Y-48

```

```

180 IF X<X1 THEN " "FARTHER WEST"
190 IF X<X1 THEN " "FARTHER EAST"
200 IF Y<Y1 THEN " "FARTHER NORTH"
210 IF Y<Y1 THEN " "FARTHER SOUTH"
212 IF X<X1 AND Y<Y1 THEN " " " " "FAR
THER NORTH WEST"
214 IF X<X1 AND Y<Y1 THEN " " " " "FAR
THER SOUTH WEST"
216 IF X<X1 AND Y<Y1 THEN " " " " "FAR
THER NORTH EAST"
217 IF X<X1 AND Y<Y1 THEN " " " " "FAR
THER SOUTH EAST"
220 IF X=X1 THEN " "X COORD IS CORRECT
"
230 IF Y=Y1 THEN " "Y COORD IS CORRECT
"
240 IF X=X1 AND Y=Y1 THEN 1000
250 IF X>9 THEN 140
260 IF Y>9 THEN 140
270 IF X<1 THEN 140
280 IF Y<1 THEN 140
290 POSITION 0,19: " #6;T;" TURNS LEFT"
291 COLOR 45:PLOT X,Y
295 FOR A=0 TO 800:NEXT A
300 NEXT T
310 " " " " " "YOU RAN OUT OF TURNS."
320 " "THE SNARK WAS AT "X1;" "Y1
325 COLOR 171:PLOT X1,Y1
330 FOR A=0 TO 1000:NEXT A
340 GOTO 5500
1000 FOR S=100 TO 1 STEP -1
1500 SOUND 0,5,10,6
2000 SOUND 1,5-1,10,6
2500 NEXT S
3000 " " " " "CONGRADULATIONS!!!"
3500 " "YOU FOUND THE SNARK IN "9-T+1
;" TURNS!"
3550 COLOR 171:PLOT X1,Y1
4000 FOR A=0 TO 500
5000 NEXT A
5001 SOUND 0,0,0,0
5500 POSITION 14,0: " #6;"PRESS"
5555 POSITION 13,1: " #6;"START"
5556 FOR QM=0 TO 255
5700 IF PEEK(53279)=6 THEN SOUND 0,0,0
,0:SOUND 1,0,0,0:CLR :RUN
5800 SETCOLOR 2,0,QM:SOUND 0,QM,10,6:5
OUND 1,QM+4,10,6
5900 NEXT QM
6000 GOTO 5556
9000 REM
9500 RETURN

```



# MANDELBROT SET

The front side contains the four programs listed within the Mandelbrot articles in this issue. They allow you to create the Mandelbrot data file, plot the data file to screen, investigate where to make the required colour changes, and finally to dump the screen to printer, colour register by colour register (for colour printouts), or in a variety of grey shades. There are two versions; one in BASIC, the other compiled using MMG compiler. To load the compiled version go to DOS and load through the L option (Binary load).

The flip side contains a slide show (using Fader 2) of graphics 7+ Mandelbrot screensaves. These include all the ones shown in this magazine plus a few more and were generated by Dick and Rita.

To generate your own Mandelbrot use Program 1 and refer to Fig.1 in the article. This is the complete set, and from here you can select the coordinates to investigate any part of that set. ACORNER (Real coordinate) is the horizontal axis, BCORNER (Imaginary coordinate) is the vertical axis. SIDE is the horizontal length of the 'square' you wish to view. (The ratio is 1 horizontal to .64 vertical.) The smaller the SIDE value, the more powerful the zooming function and you find yourself deeper within the set. I have gone to 8 decimal places (and further). The deeper you go, the more precise you need to be with your measurements to get the right coordinates to find something of interest, and you do need some luck, otherwise you may find nothing but the blackness of the set itself.

Load in program 1. Insert a disk with 250 free sectors, this will be your data disk. Follow the prompts. Turn off the screen and leave the computer and drive to work for the next 10-30 hours while you do all those other tasks needing attention. When all the computation has finished and the file has been completed, the screen will show FINISHED and the name of the file it was saved to.

Load in Program 2. Type in the name of the picture file, then the name of the data file to generate the picture. For the first run through select option 2 (MOD 3) and watch the mystery of the selected area unfold before your eyes (15 minutes). Save this screen by pressing the SELECT KEY. Use Program 5 (Loadscrn) to retrieve this picture at a later time.

Program 3. Follow the prompts. This program shows you at what levels of iterations activity was occurring. This will help in the choice of where to set the colour changes for the best effects. After noting (or dumping) the information generated by this program use that information to experiment where to place your colour changes in program 2. Continue experimenting until you have the effect you desire.

Program 4. If you have a PX-80 printer this will dump your screen to the printer either in shades of grey, or in separate colour registers.

Program 5. This is a rough and ready retrieval program. Follow the prompts (graphics 15 is graphics 7+). I added this as an afterthought, just in case you did not have the means to retrieve the picture files you had created. This small program will actually retrieve any type of saved screen except for compressed screens. More on screensaves and retrievals in another issue.

PROGRAM 1 MANDELZOOM part 1 - MANDLCAL.BAS  
PROGRAM 2 MANDELZOOM part 2 - MANDPLOT.BAS  
PROGRAM 3 DATACHK.BAS  
PROGRAM 4 COLDUMP.BAS  
PROGRAM 5 LOADSCRN.BAS

ALSO: Programs 1, 2 and 3 have been compiled. To run the compiled version go to DOS and load through L option. The filenames are the same, but the extenders are COM.

SIDE 2 is an autorun. (Press START to speed up the display of each screen.) Just slip it into the drive, turn on the computer, sit back and watch a slide show of what you can produce using the programs on the front side of the disk.

— Rita

## MANDELBROT SETS

and the ATARI

XE by DICK KELLETT (reprinted from M.A.C.E. Australia, March '86)

Have you looked at those computer generated pictures in SCIENTIFIC AMERICAN AUGUST 1985 and YOUR COMPUTER YEAR BOOK, JANUARY 1986 and thought you might like to try the same type of graphics? The articles in these magazines suggest setting up arrays of up to 1000 by 1000 and using 1000 iterations to check if each point is in the MANDELBROT set. This is fine if you have access to a super mini or a mainframe but not much help with only 32K available. While the ATARI cannot match the resolution of the published pictures, GRAPHICS 7+ (GRAPHICS 15+16 on XL and XE models) will give sufficient resolution and three colours plus background colour to produce interesting pictures.

The pictures are produced by using the equation  $Z \pm 2 + C$  where Z and C are complex numbers and repeating the calculation with the answer replacing Z in the equation. Counting the number of iterations before  $Z = 2$  and assigning a colour to this number generates the picture. If the number of iterations exceeds the selected maximum (in our case 100 iterations) the area (pixel) is in the MANDELBROT set and is plotted in the background colour.

After many frustrating hours plotting screens directly and having a single colour or a small area in one corner, I realized the number of iterations selected to change colours was very important and differed for each picture.

My approach to the problem is to use two programs. The first one called MANDLCAL.BAS selects an area according to the formulae in the SCIENTIFIC AMERICAN article, scales it to suit a GRAPHICS 7+ screen and stores the results on disk as a DATA file of 246 sectors. I used one hundred iterations to approximate whether or not the point was in the MANDELBROT set.

The second program called MANDPLOT.BAS allows you to select the levels of iteration which colour changes will occur at, and then plot the point in the selected colour. Points in the MANDELBROT set are always plotted in the background colour.

The colour changes may be selected to change at preset counts, plot through the colours in MOD.3 or plot colours below the first change in MOD.3.

The same colour is used for the lowest and highest counts as there is sufficient difference in the position of the points plotted to avoid running the areas together. It almost gives the effect of having an extra colour.

Both of these programs run very slowly. MANDLCAL takes between ten and twenty hours to calculate the data file. The worst case involves  $100 \times 159 \times 191$  separate calculations for an area completely in the MANDELBROT set. Obviously, an area with a lot of background points takes longer to calculate than one with a lot of colour.

MANDPLOT will set up a graphics 7+ screen and plot the picture from the data file in approximately twenty minutes. Before the picture is plotted you will be asked for a file name to save the picture to. If you do not select a name, the picture will be saved using the default name of PICTURE.

The pictures are saved as 62 sector files. I prefer to save my pictures with individual names and use the DOS copy file option to create a new file called PICTURE, which can then be loaded into ATARI ARTIST (MICRO ILLUSTRATOR) by pressing the CLEAR key after ATARI ARTIST has been loaded. The colours can then be adjusted and pattern fills added as required. The picture can then be saved in the normal way and used with FADER 11 as a slide show. The pictures could also be put through Rapid Graphics Converter and be used as a background file for MOVIE MAKER etc.

Both these programs will compile with the MMG compiler. MANDLCAL does not show a significant increase in speed and I let the program run overnight and while at work the next day. When compiled MANDPLOT will plot the picture in approximately ten minutes instead of twentyfive minutes.

If you try these programs I suggest you obtain a copy of the photograph of the full MANDELBROT set from the SCIENTIFIC AMERICAN article, as it shows the co-ordinates for the set. This will allow you to select the starting points (ACORNER, BCORNER) and the length of the side for the area to be plotted (SIDE).

(postsript - use Fig.1 after Rita's article)

To get started try the following points. The first set of figures should give you the complete MANDELBROT set (a colour version of Fig.1).

ACORNER=-2.5 BCORNER=-1.25 SIDE=3.5  
ACORNER=-.2665 BCORNER=-.0049 SIDE=.002  
ACORNER=-.9 BCORNER=.263 SIDE=.005

## ST VT52 COLORS

(Reprint: Page 6, Caltari)

Tired of looking at plain old black and white on your VT52 emulator? I've managed to dig up a few things you can do to give greater cursor and color control while telecommunicating. First, go to Set RS-232 and select half-duplex (so you can see what you're doing).

Next press Esc followed by B. This will move the cursor down one line. After mastering that, try these (each preceded by the Esc key):

A Cursor Up	b# Select character color #
C Cursor Right	c# Select background color #
D Cursor Left	e, f Enable, Disable cursor
E Clear Home	j Save cursor position
H Cursor Home	k Move to position
J Clear Below Cursor	p, q Reverse on, off
L, M Insert, Delete Line	(more than these exist!)

— Craig Maynard



## Delays in Action!

In the past I've used 'waste time' loops to provide delays in Action! programs. I also like to use this time for other program functions. Software timers built into the Atari provide a means of doing this. I will give examples of three different ways of providing delays and continuing other program functions at the same time.

### Poll Flags

There are five software timers which decrement 16 bit counters, once each vertical blank interrupt. Three of these (#s 3,4 & 5) set a flag location to zero whenever they themselves reach zero. 'Pollflag' gives an example of using system timer #3. A machine language routine is necessary to set the timer without conflicts. This routine is located at \$E45C (58460). It requires the timer number to be in the accumulator, the high byte of the time interval in the X register and the low byte in the Y register. Fortunately this is easy to do in Action! A PROC is defined to start at location 58460 and passed the proper parameters. Notice the code already exists so the PROC needs no body or RETURN statement. Timer #3 sets a flag at \$022A upon time-out. The variable 'flag' is set to this location. The flag is set to 1, the timer is set and then it is just a matter of checking every once in a while to see if flag has returned to zero. With 16 bits you can get times anywhere from 1/60th of a second to about 18 minutes.

### Interrupt at timeout

Instead of setting a flag, timers 1 & 2 jump to a given routine when they time out. Timer 1 is used by the OS and should probably be left alone. 'Inttime()' gives an example of using timer #2. Before setting the timer you put the address of the routine to jump to into locations \$0228, \$0229 (PokeC(\$0228,timeup)). In this case the called routine just changes the background color. More complicated things could be done but don't try doing any I/O. Remember you are in the middle of a VBI; I/O will do strange things here.

### Many timers

Suppose you need more timers than are available in the OS. Simple, create your own. Let's stick to single byte timers for simplicity. We need locations to hold the counts (cntdn) and locations for flags (the array cntflag). We also need to insert our own code into the VBI routine to decrement the cntdn locations.

The same machine language routine at \$E45C is used to insert our routine. The 6 in Setvect(6,...) places it in the immediate portion of the VBI. Notice the in-line code at the end of Vblank() to return to the OS portion.

Rotate() checks each of the flags and updates the appropriate character when a timeout occurs (cntflag(j)=0). The time is then reset by putting a number in cntdn(j) and setting cntflag(j) to one. The time interval to use is taken from maxcnt(j) and every so often this number is increased causing the spinning to slow down. If the number reaches 15 the updating is skipped and the character remains stationary. In the main routine there is a long random interval after which one of the maxcnt's is set to 1, causing that one to spin rapidly again. Notice the WHILE statement in Rotate(). I originally had this an IF... THEN statement. This made the DO OD loop infinite and it took me a while before I realized what was going on. Something to watch out for in the future! I also made the mistake of having the counters for each routine (i,j & k) all the same (j), and made that 'j' global. They could all be the same (j) if they were defined separately in each routine, but making j global meant that the counter in one routine affected the counter in another. This also took a while to debug and is something else to watch out for. I'm learning to stay away from global variables if possible.

I hope these examples will help you in another approach to delay routines in Action!

— Stan Ockers

## SOLAPAK

This is the first column I have written for the newsletter, I hope it is not the last. The ST is my first computer, I came to it as a novice with a lot to learn. The program I was given to review is from Solar Powered Software (1807 N. Evergreen, Chandler, AZ 85224), **SOLAPAK** (\$30), and contains a spooler, an adjustable ram disk and a screen saver program. Documentation for these programs is contained on the disk.

The instruction manual suggests that you start out with a printed copy of Readme.Doc to gain basic information on how to install these programs. I found out later that you also must have a copy of Defaults.Def and Spooler. Doc before you can get any where.

For the novice, I find the directions to be ambiguous. I get nervous trying to figure out what parameters or files are best suited to install for my printer. One of the first things to be done is to "choose" the format needed to set up your printer. I have an Epson compatible Pro Writer Junior (C.itoh) printer and was not sure which file to use. The company does supply formats for 8 different printers, two Epson and one C.itoh. Some experimentation shows the file for the Epson FX85 printer was very close.

## 14

The next thing to do is create an Auto folder and copy two files to this on your boot disk. I did not copy these files into the auto folder, but onto the root directory. When I booted up the program, I found I had indeed erred and simply reread the directions and copied these files into the Auto folder.

After I did this, I installed the screen saver and spooler as a desk top accessory — you do have the option to use the ram disk and spooler as a separate program from a disk. I was not sure of the benefit of the screen saver when I first started. I thought that it might just be one of those nice things to have around and never use. After seeing it work, I am convinced it can be quite beneficial, especially if there are young kids in the household who use the computer. It does reduce the worry that the kids will leave the monitor on when they are through. You can take a short trip to the facilities, or answer the phone without worry as well.

The ram disk and spooler seem to work as advertised. I have found some major inconveniences in their use, one is that you have to do a lot of work to change the size of the disk, disable it, or disable the spooler. To access the proper documentation, you must first load in the ST Writer program, unless you just want to type it to the screen. You might find it helpful to print all the document files for reference.

This was my first chance to use a spooler, and I was surprised by the ease of use and the fact that you can actually use the keyboard while printing a document. You can have up to 32 files in memory to print out as needed (520 ST with ROMs installed). I was distressed to learn I cannot use the spooler with First Word. When printed, there were no spaces between the words. Formatting information used by the printer driver in your word processor does not seem to get through this spooler. Every thing else seemed fine both in draft and Near letter quality. I have tried to print out files from ST Writer as well. I found that it printed the numerical information at the top of the screen as well as the text, and the text was printed as shown in the file — that is, not properly formatted. You will need to print ST Writer files to the disk, and then spool it. The disk file will be properly formatted.

There are some draw backs in the useage of the spooler and sizing the ramdisk, but for the price tag, the security of the screen saver could be quite worth the investment.

— Mike Rogers

## LITTLE ALARMS

by A.J. Aspromatis

One of the functions of a newsletter is the dissemination of information from and for the benefit of its members and interested parties. This information takes many forms and "the warning" is one of the most valuable. One might say the subtitle of this column might be "Let the Buyer Beware!"

First, in regard to the DataSoft announcement in the March A.C.E. for programmers to submit their wares, a warning. Several months ago this programmer sent them a proposal with demo disk of VIDEO GAMESHOW, a TV gameshow simulation being worked on at the time. Included were a return envelope with postage and a reply-postcard. After waiting a fair amount of time without acknowledgement, a follow-up letter sent, requesting the status of their review of the submission. With the passing of more than 5 months, no reply and no returned software. 'Nuff said?

Next, for the BASIC programmer who is tired of speeding up program execution with kludges or limited instruction compilers, C! Among other reasons for adding C to one's language repertoire is for the learning of structured programming techniques, which result in more elegant programs and easier debugging. Ralph Walden's "ACE-C" C compiler even removes the obstacle of cost from before the ranks of the timid. It is available from the A.C.E. program library for the nominal price of \$10. A truly unsurpassed value! (Beginners should also obtain a book on C such as "The C Primer" by Hancock & Krieger and a text editor such as the PROGRAM-TEXT EDITOR formerly available from APX.) This programmer used C as a utilities development tool to assist in such mundane tasks as text compression and disk protection for VIDEO GAMESHOW. BASIC would have been too slow (as a side note, ACE-C allows inclusion of machine code for even faster operation).

Aside from the few operational quirks of ACE-C, there is, however, a bug in one of the routines of the runtime file ENGINE.OBJ, specifically for the function "ftoi()" (Floating point TO Integer). The problem is the hi/lo bytes of the integer number are returned to the calling program in reversed order. The C program listed here, when compiled and run, will fix the offending code. I wish all problems in life were so easily resolved.



# ATARI OCTOPUS

There we were, me and my new XM301 modem, 30 fathoms deep into the newest local bulletin board, looking eyeball to boob tube into the secret expert user codes for the system. Oh no! My disks are already full of booty and I don't have any formatted floppies left. How can this valuable information be brought back to the surface? Desperately, I switch on my trusty MX80 printer and try to execute a screen dump. Oh no! Nothing happens! the system access time is quickly running down. My screen echos "One minute left". Time ticks away. Still no printer response!!! My screen echos "Thankyou for using the Best Little Bulletin Board in Town, ya All come back real soon now." Rats! I'll never figure out how to get back to where the secret expert user codes live!

Back on the surface, the problem becomes clearly evident. The new Atari XM301 modem is really a great buy for less than \$39. However, for Atari to be able to produce this thing to sell at that price, something had to be left off. What they left off was the serial bus daisy chain connector (probably saved them a buck and change). That's the connector which lets you plug one disk drive into the next, into a printer and so on. The only other Atari peripheral with the daisy chain port problem is the cassette tape drive. Atari probably figures no one needs cassettes any more, so why not let the XM301 be the last peripheral in the chain?

Well, you see, I have another great buy. An MPP alias CARDCO parallel printer interface which is also missing the daisy chain connector. The result is I can't have my printer and modem plugged in at the same time. It gets to be a real pain untangling and sorting cables every time I want to do something. I pull out a disk drive to switch connections and three other cables pop off at the same time (of course I don't notice until I put everything back). Modem programs won't boot until I plug the modem in; Printshop gets mad if the printer isn't plugged in. Fickled (sic) software!

To put an end to my problems I built an octopus for my Atari. Essentially, this means I cut two Atari serial interface cables in half, and spliced the same color wires together inside a junction box. I ended up with one connector to plug into the computer, and three others which can plug into disk drives, printers, or whatever. This frees up to three serial ports for single ended peripherals. This modification also helps to untangle the nest of cables around the Atari.

Being a graduate of a local junior college (after 13 years of intensive study), this was no big problem to design and build. Anyone who has mastered the art of fabricating miniature thermonuclear devices should be able to do it also, particularly if they follow the accompanying instructions.

Hopefully, you already have two serial bus cables. Try to use only Atari cables since they have the greatest chance of being wired with the same color code (you need some luck here). The only thing you really need to buy is a small metal box to use for the junction box and perhaps a small screw terminal strip, with 15 or more connectors, in the event you're not an ace solderer. I recommend a metal box over a plastic one to prevent emi (wavey lines on your tube) when using the peripherals.

Disassemble the connectors at one end of each of the two cables by removing the screw and nut. Take off the end shell and check the wire color codes to make sure the same color wire goes to the same pin on each connector. If they don't, try another cable, or if you're really brave (e.g., nuts), make a diagram to translate the colors. Once you're sure, you can reassemble the cable ends.

Now comes the fun part. As you remind yourself these cables will cost at least \$15 a piece to replace, if you can find them, cut each cable into two pieces. You may want to make the cut so each cable is a different length as required by your individual computer setup.

With a good pair of wire strippers or a sharp knife, strip the black outer insulation back about three inches. **Be very careful not to cut into the inner wires.** Then strip the individual wires back about 3/4 of an inch.

Drill holes for the cables in a box and pass the wires through the holes before you begin making the connections. If you don't do this first, you will get very upset when you have to unsolder and resolder everything.

Join each set of the same color wire combination (four wires) by twisting them together and soldering or connecting them together at a connector terminal. When you finish, you will notice some bare wires (the shield) left over. Connect these together and fasten them to the metal box.

If you soldered the wires, insulate the exposed connections with electrical tape or heat shrink tubing. This will help prevent the need for a fire extinguisher when you plug things back in.

Triple check everything carefully, short circuits are expensive. Place the cover on the box and you're ready to go. Just plug one end of your octopus into your computer, and the other three ends can connect to your other peripherals. Your single ended peripherals can then be connected to open daisy chain connectors on the other peripherals.

— Ron Robinson

## POWER

(Reprint: January, 1986 issue of STATUS) If you've ever had a power supply go bad (as mine did) you know how long it can take to get a replacement. A few phone calls to local vendors reveals a one week wait at the minimum. Undaunted (and impatient), I decided to build my own power supply.

By using the circuit diagram (See Figure 1) and following the construction notes which follow, you can have a "beefy" power supply which will handle any demand the 800XL can place on it. And yes, it will indeed power a RamDisk modification. As an added feature, I've included surge protection with the power supply.

The heart of this power supply is a LM323K +5 volt regulator IC. With the proper heat sink, the regulator can provide up to 3 amps to your computer. The 800 XL requires less than 1.5 amps (the 130 XE less than 2 amps), so with a 3 amp rating this regulator is ideal for the power supply.

For connecting to your computer, you may purchase a 7-pin "DIN" connector or you may cut the cable from your old supply. If you look at the diagram (see Figure 3), you'll see pins 1, 4 and 6 are the +5 volt output. Pins 3, 5 and 7 are the ground connections. With an Ohmmeter, find the wire which you cut connecting to pins 1, 4 and 6 and tag it with "+5" for future use. The other wire is the ground and it connects to pins 3, 5 and 7.

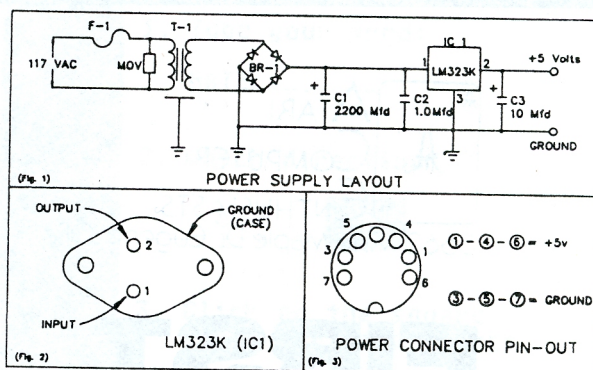
Follow the schematic (see Figure 1) and wire your project carefully. Observe the polarity of the electrolytic capacitors and study the pin-out configuration of the LM323K (see Figure 2) before making the connections.

Connect the Metal Oxide Varistor (as shown in the diagram) for surge protection. Connect the black and white wires of the line cord to the input of the transformer. A fuse holder should be wired to one of the inputs of the transformer. The green wire (if you are using a three wire cord) should be connected to the metal case of the transformer.

One final construction note. The LM323K must be mounted on a heat sink. I chose a plastic project case with a metal top (see Parts List) and mounted the LM323K to the metal top with heat sink compound. As an option, you can mount the regulator on a heat sink designed for a "TO-3" transistor case. I also recommend a TO-3 socket for the regulator for easy installation and removal.

Before connecting the power supply to your computer, power up your unit and check for +5 volts at the output. If all seems well than try your computer. If you notice any "ripple" on the screen, then recheck all your solder connections. It may be necessary to connect the case of the 12 volt transformer to ground. In extreme cases, you may have to shield the entire transformer with a piece of metal connected to ground.

Good luck with your project!



### Power Supply Parts List

- T1 - 12 volt transformer (Radio Shack 273-1352 or equivalent);
- BR1 - Bridge rectifier (RS 276-1146 or equiv.);
- IC1 - LM323K (available at local electronics suppliers);
- C1 - 2200 mfd electrolytic capacitor (RS 272-1020 or equiv.);
- C2 - 1.0 mfd capacitor (RS 272-1055 or equiv.);
- C3 - 10 mfd electrolytic capacitor (RS 272-1013 or equiv.);
- MOV - Metal Oxide Varistor (RS 276-571 or equiv.);
- Misc - Project enclosure (RS 270-232 or equiv.);
- Fuse holder.



## Atari Computer Enthusiasts

The ACE Newsletter is the first Atari user group newsletter to be published anywhere in the world. We began publishing in April, 1981.

A.C.E. is an independent, non-profit and tax exempt computer club and user group with no connection to Atari Corp. We are a group interested in educating members in the use of Atari computers and publishing the latest News, Reviews and Rumors.

**All our articles, reviews and programs come from you, our members.**

Our membership is world wide. Membership fees include the A.C.E. newsletter (10 issues per year) and high access in the A.C.E. Bulletin Board System. Dues are \$14 a year for U.S., and \$24 a year overseas airmail.

**Subscription Dept.:** 3662 Vine Maple Dr., Eugene, OR 97405

**President** — Dick Barkley, 2907 Wingate, Eugene, OR 97405  
(503) 344-5843

**Vice President** — Larry Gold, 1927 McLean Blvd., Eugene, OR 97405  
(503) 686-2490

**8-Bit Librarian** — Chuck & Jody Ross, 2222 Ironwood, Eugene, OR 97401

**ST Librarian** — Jim Bumpas, 4405 Dillard Road, Eugene, OR 97405  
(503) 484-4746

**Editors** — Mike Dunn, 3662 Vine Maple Dr., Eugene, OR 97405  
(503) 344-6193

Jim Bumpas, 4405 Dillard Road, Eugene, OR 97405  
(503) 484-4746

Larry Gold, 1927 McLean Blvd., Eugene, OR 97405  
(503) 686-1490

**E.R.A.C.E.** (Education SIG Editor) — Nora Young, 105 Hansen Lane  
Eugene, OR 97404 / (503) 688-1458

Send 50c stamps or coin (\$1 overseas) to the Librarian for updated ACE Library lists.

**A.C.E. Bulletin Board**  
(503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a 130 XE upgraded to 320k by Charlie Andrews, 2 double-density disk drives, and a 10-meg hard disk from Supra Corp, an Epson MX80 printer, a 2400 US Robotics Courier modem, running the PD/M Bulletin Board software.

### Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct. 1981 to June, 1982.

Volume 2 contains July 1982 through June 1983.

Only \$12 each (\$2 extra for airmail). Available only from George Suetsugu, 45-602 Apuapu St., Kaneohe, HI 96744.

## TYPESETTING FROM YOUR COMPUTER

**ATARI OWNERS:** If you have a modem, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by **TYPESETTING** your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." And it's easy to encode your copy with the few typesetting commands you need.

### COMPLETE CONFIDENTIALITY GUARANTEED

— Bonded for your protection —

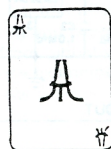
**PUBLICATION DESIGN, EDITING, & PRODUCTION**

## Editing & Design Services

**30 East 13th Avenue Eugene, Oregon 97401**  
**Phone 503/683-2657**

### SortFinder Index

A composite index of Atari related articles in four popular computer periodicals, including the ACE Newsletter. Volume 1 covers April, 1981 through June, 1983. Volume 2 covers July, 1983 through December, 1985. Only \$6 per printed copy or \$11 per disk copy for ACE members. Available from: Jim Carr, 2660 S.W. DeArmond, Corvallis, OR 97333.



ATARI  
COMPUTER  
ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

# FIRST CLASS MAIL